Hr

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/632,224	07/31/2003	Thomas McDonald	CNTR.2141	2078

23669 7590 02/22/2006

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER	
MOLL, JESSE R	

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 02/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/632,224	MCDONALD, THOMAS	
	Examiner	Art Unit	
	Jesse R. Moll	2181	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 October 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4 and 7-32* is/are rejected.

7)☒ Claim(s) *5 and 6* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 July 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *20 October 2005*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☒ Other: *See Continuation Sheet*.

Continuation of Attachment(s) 6). Other: IDS: 28 September 2005, 7 July 2005, 31 July 2003.

DETAILED ACTION

1. Claims 1-32 have been examined.

Acknowledgment of papers filed: oath, specification, drawings, and IDS, on July 31, 2003, IDS on July 11 2003, IDS on September 28, 2003, and IDS on October 20, 2003. The papers filed have been placed on record.

Priority

2. Applicant's claim for the benefit of a prior-filed application under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, or 365(c) is acknowledged. Applicant has not complied with one or more conditions for receiving the benefit of an earlier filing date under 35 U.S.C. 119(e) as follows:

The later-filed application must be an application for a patent for an invention which is also disclosed in the prior application (the parent or original nonprovisional application or provisional application). The disclosure of the invention in the parent application and in the later-filed application must be sufficient to comply with the requirements of the first paragraph of 35 U.S.C. 112. See *Transco Products, Inc. v. Performance Contracting, Inc.*, 38 F.3d 551, 32 USPQ2d 1077 (Fed. Cir. 1994).

The disclosure of the prior-filed application, Application No. 60/440,063, fails to provide adequate support or enablement in the manner provided by the first paragraph of 35 U.S.C. 112 for one or more claims of this application. Accordingly, claims 1-32

are not entitled to the benefit of the prior application because the specification of the prior application does not mention a main aspect of the invention (the kill queue). The prior application does mention using a kill bit, but does not go into enough detail about how the kill bit is used in a queue to enable the claims.

Claim Rejections - 35 USC § 101

3. Claim 32 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 32 is not limited to tangible embodiments. In view of Applicant's disclosure, specification page 44, lines 5-11, the medium is not limited to tangible embodiments, instead being defined as including both tangible embodiments (e.g., semiconductor memory, magnetic disk, optical disk) and intangible embodiments (e.g., carrier wave). As such, the claim is not limited to statutory subject matter and is therefore non-statutory. Examiner requests that claim 32 read "A program embodied on a computer readable medium, comprising..." and that the specification be changed to remove carrier wave as possible computer readable medium.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371 (c) of this title before the invention thereof by the applicant for patent.

5. Claims 1-3, 8-13, 15, 17-21, and 29-32 are rejected under 35 U.S.C. 102(e) as being anticipated by Roth et al. (U.S. Patent No. 6,754,808 B1), herein referred to as Roth et al.

6. Referring to claim 1, Roth et al. discloses an apparatus for killing an instruction loaded into an instruction queue (instruction pipeline stages DEC-EX3; fig. 4. Refs. 402-408; col. 3, lines 38-44; Note that Examiner considers an instruction pipeline to be an instruction queue. A queue is a data storage device which outputs data in the same order in which it was entered. An in-order instruction pipeline stores instruction while they are being executed, and go through the pipeline in order and exit) of a microprocessor during a first clock cycle (fig. 4, clock cycle 3; instruction enters the DEC stage of the pipeline) and output from a bottom entry (EX3 stage of the pipeline) of the instruction queue during a second clock cycle (instruction exits EX3 and enters WB stage), subsequent to the first clock cycle (instruction is written back after it is decoded), the apparatus comprising: a kill signal (signal produced by fig. 5, ref. 506; col. 4, lines 19-21), for conveying a value generated during a third clock cycle (kill signal generated in a stage of the pipeline; fig 6, KILL.DEC, KILL.AC and kill signals in stages EX1-WB; col. 6, lines 3-5) subsequent to the first clock cycle (kill signals can be generated after the DEC stage; see col. 6, lines 3-5); a kill queue (fig. 5, refs. 504, 510, 514, & similar items in stages DEC-WB; col. 4, lines 49-57), coupled to said kill signal (col. 4, lines 16-

21), for loading said kill signal value generated during said third clock cycle (col. 4, lines 17-21), and for outputting said kill signal value during the second clock cycle (col. 4, lines 3-6 (Examiner considers the signal to be output by items 504, 510, and 514. If a kill signal is sent to valid bit qualifier 504, the instruction valid signal [which is functionally equivalent to the kill signal] will travel down the pipeline through latch 510 and valid bit qualifier 514. Note that the valid bit [kill signal value] is output to the write back logic. The write back logic then uses the information to determine if the instruction is committed) and a valid signal (signal produced by valid bit qualifier in the WB stage), coupled to said kill queue, generated during the second clock cycle (clock cycle when the instruction enters the WB stage) for indicating whether the instruction is to be executed by the microprocessor (col. 4, lines 3-6), wherein said valid signal is false if said kill signal value output by said kill queue during the second clock cycle is true (col. 4 lines 16-26; Note that Roth et al. discloses the use of a valid queue which is analogous to the claimed kill queue. However, Roth et al. uses a false value to represent that an instruction is to be killed. This difference makes no difference functionally and is only an alternative bit representation)

Further note that claims 17 and 32 recite the same limitations as claim 1 but are claimed as a method and computer readable medium. The method of claim 17 is disclosed by the method used by the invention disclosed by Roth et al. The medium of claim 32 is disclosed by any software executed on the invention disclosed by Roth et al.

7. Regarding claim 2, Roth et al. discloses the apparatus of claim 1, wherein said third clock cycle is a same clock cycle as the second clock cycle (col. 6, lines 3-5).

Note that instructions can be killed in any of the stages. If an instruction is killed in the write back stage, it enters the queue (valid bit qualifier of the write back stage) and leaves in the same clock cycle.

Further note that claim 18 recites the same limitations and is therefore rejected using the same reasons.

8. Regarding claim 3, Roth et al. discloses the apparatus of claim 1, wherein said third clock cycle is a clock cycle prior to the second clock cycle (the instruction can be killed in any stage of the pipeline; col. 6, lines 3-5; Note that if an instruction is killed in any stage before the WB stage, then the kill signal is generated before the instruction exits the queue).

Further note that claim 19 recites the same limitations and is therefore rejected using the same reasons.

9. Regarding claim 8, Roth et al. discloses the apparatus of claim 1, wherein said kill queue comprises: a plurality of entries (fig. 5, ref. 510 & latches between all other stages), for storing a plurality of values of said kill signal (col. 4, lines 49-51) generated during a corresponding plurality of clock cycles (fig. 4, kill signal generated for instructions: i, i + 1, i + 2, etc.).

10. Regarding claim 9, Roth et al. discloses the apparatus of claim 8, wherein each of said plurality of kill queue entries comprises a load data input (col. 4, lines 17-21), coupled to receive said kill signal (col. 4, lines 51-54).

11. Regarding claim 10, Roth et al. discloses the apparatus of claim 8, wherein each of said plurality of kill queue entries comprises a hold data input, coupled to receive a current value of said entry (col. 5, lines 36-38; Note that latches require a the output of the latch to be input into itself to function (see The D Latch, connection between Q and the input of the bottom right NAND gate).

12. Regarding claim 11, Roth et al. discloses the apparatus of claim 8, wherein each of said plurality of kill queue entries comprises a shift data input, coupled to receive one of said plurality of values of said kill signal from one of said plurality of entries above said each of said plurality of kill queue entries (fig. 5, connection between refs. 504 and 510; col. 4, lines 62-67).

13. Regarding claim 12, Roth et al. discloses the apparatus of claim 8, wherein the instruction queue comprises a plurality of entries for storing a plurality of instructions (col. 3, lines 40-44), wherein said plurality of kill queue entries store corresponding said kill signal values for said plurality of instructions stored in said plurality of instruction queue entries (col. 4, lines 7-10).

14. Regarding claim 13, Roth et al. discloses the apparatus of claim 1, wherein the instruction comprises a variable length instruction (col. 5, lines 12-16).

15. Regarding claim 15, Roth et al. discloses the apparatus of claim 13, wherein the instruction is provided to the instruction queue (instruction passed from alignment stage to the decode stage; fig, 6, ref. 602; Note that in the alignment stage, processors receive instruction from cache/memory and format them into individual instructions to be decoded) during the first clock cycle (instruction entering the queue (DEC stage of the pipeline)) by an instruction formatter (the hardware in the align stage), said instruction formatter determining a length of the instruction (col. 5, lines 1-3).

16. Regarding claim 20, Roth et al. discloses the method of claim 17, further comprising: formatting said instruction prior to said loading said instruction into said first queue (alignment stage formats the instruction; see above regarding claim 15).

17. Regarding claim 21, Roth et al. discloses the method of claim 17, further comprising: determining whether said instruction is shifted down in said first queue (fig. 6 refs. 614; col. 5, lines 40—57; Note that if stall signal is true and the kill signal is not true, then the instruction has been stalled) after said loading said instruction into said first queue (the AC stage occurs after the instruction enters the first queue); and shifting down said value of said kill signal in said second queue after said loading a value of said kill signal into a second queue (the AC stage can occur after the second clock

cycle; see above regarding claim 1), if said instruction is shifted down in said first queue (col. 5, lines 40--57).

18. Referring to claim 29, Roth et al. discloses a microprocessor, comprising: a first queue (instruction pipeline stages DEC-EX3; see above regarding claim 1), for receiving an instruction for buffering therein (instruction entering the DEC stage; see above regarding claim 1); logic, coupled to said first queue, for detecting a condition wherein said instruction must not be executed by the microprocessor (kill signal generated during one of clock cycles 3-8; fig 6, KILL.DEC, KILL.AC and kill signals in stages EX1-WB; col. 6, lines 3-5; note that the logic must exist to generate the kill signal since it is generate), wherein said logic generates a true value on a signal to indicate said conditions wherein said true signal value is generated subsequent to said instruction being received by said first queue (kill signal can be generated while in any stage of the pipeline; see above regarding claim 1); and a second queue (fig. 5, refs. 504, 510, 514, & similar items in stages DEC-WB), coupled to said logic, for loading said true signal value and subsequently outputting said true signal value contemporaneously with said first queue outputting said instruction (send instruction and kill signal to write back stage; see above regarding claim 1), wherein the microprocessor invalidates said instruction in response to said true signal value and does not execute said instruction (if the instruction is not valid, the instruction is not completed; see above regarding claim 1).

19. Regarding claim 30, Roth et al. discloses the microprocessor of claim 29, wherein said second queue comprises: a plurality of storage elements (fig. 5, ref. 510 & latches between all other stages), for storing a plurality of values of said signal generated by said logic (col. 4, lines 49-51) during a corresponding plurality of clock cycles (col. 4, lines 7-10).

20. Regarding claim 31, Roth et al. discloses the microprocessor of claim 30, wherein said first queue comprises a plurality of storage elements for storing a plurality of instructions (col. 3, lines 40-44), wherein said plurality of second queue storage elements store corresponding said signal values for said plurality of instructions stored in said plurality of first queue storage elements (col. 4, lines 7-10).

Claim Rejections - 35 USC § 103

21. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22. Claims 4 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kroesche (U.S. Patent No. 6,968,444 B1) in view of Roth et al. (U.S. Patent No. 6,754,808 B1).

23. Regarding claim 4, Kroesche discloses an apparatus for killing an instruction loaded into an instruction queue (fig. 2, queue 210; instruction is loaded from decoders; col. 6, lines 11-13; col. 6, lines 42-43) of a microprocessor during a first clock cycle (decode clock cycle) and output from a bottom entry of the instruction queue during a second clock cycle (clock cycle when instruction is outputted to issue position 0, 1, or 2; col. 6, lines 54-56), subsequent to the first clock cycle (when the queue is not empty, the instruction will sit in the queue for at least 1 clock cycle; col. 6, 35-38) and a load signal (pointer; col. 7, lines 35-40) for indicating during the second clock cycle (when instruction is retired from the queue) whether the instruction was loaded into the bottom entry of the instruction queue during the first clock cycle (If the instruction was loaded into the bottom of the queue, the pointer will point to the instruction).

Kroesche does not explicitly disclose a kill signal, for conveying a value generated during a third clock cycle subsequent to the first clock cycle; a kill queue, coupled to said kill signal, for loading said kill signal value generated during said third clock cycle, and for outputting said kill signal value during the second clock cycle; and a valid signal, coupled to said kill queue, generated during the second clock cycle for indicating whether the instruction is to be executed by the microprocessor, wherein said valid signal is false if said kill signal value output by said kill queue during the second clock cycle is true or that the signal is coupled to said kill queue.

Roth et al. teaches using a kill signal (signal produced by fig. 5, ref. 506; col. 4, lines 19-21), for conveying a value generated during a third clock cycle subsequent to the first clock cycle (kill signal generated during one of clock cycles after the decode

cycle; fig 6, KILL.AC and kill signals in stages EX1-WB; col. 6, lines 3-5); a kill queue (fig. 5, refs. 504, 510, 514, & similar items in all stages after the decode stage; col. 4, lines 49-57), coupled to said kill signal (col. 4, lines 16-21), for loading said kill signal value generated during said third clock cycle (col. 4, lines 17-21), and for outputting said kill signal value during the second clock cycle (col. 4, lines 64-67; passing valid bit to next [execute] stage); and a valid signal (signal produced by valid bit qualifier), coupled to said kill queue, generated during the second clock cycle (when the instruction is passed from the align stage to the execute stage) for indicating whether the instruction is to be executed by the microprocessor (col. 4, lines 3-6; writing back an instruction is considered to be executing that instruction), wherein said valid signal is false if said kill signal value output by said kill queue during the second clock cycle is true (col. 4 lines 16-26; see above regarding claim 1). Roth et al. also teaches coupling the load signal to said kill queue (col. 3, lines 60-64; a pointer in a buffer would point to both the instruction and the valid bit).

The combination of Roth et al. and would have been successful because adding a valid bit to all instructions in the pipeline to allow for instructions to be easily completed and would not drastically change the overall structure of Roth et al.

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify the invention of Kroesche by using a kill signal for conveying a value generated during a third clock cycle subsequent to the first clock cycle, as taught by Roth et al., in order to invalidate instructions which must be invalidated (Roth et al., col. 3, lines 53-58); a kill queue, coupled to said kill signal, for loading said kill signal value

generated during said third clock cycle, as taught by Roth et al., in order to preserve processor resources and improve performance (Roth et al., col. 4, lines 1-3), and for outputting said kill signal value during the second clock cycle, as taught by Roth et al., in order to make sure invalid instructions are not completed (Roth et al. col. 4, lines 3-6); and a valid signal, coupled to said kill queue, generated during the second clock cycle for indicating whether the instruction is to be executed by the microprocessor, wherein said valid signal is false if said kill signal value output by said kill queue during the second clock cycle (kill signal is the same as the valid signal) is true and that the signal is coupled to said kill queue, as taught by Roth et al., in order to keep track of invalid instructions (Roth et al., col. 3, lines 45-56).

24. Regarding claim 7, Roth et al. discloses the apparatus of claim 4.

Kroesche does not expressly disclose logic, coupled to said kill queue, for generating said valid signal during the second clock cycle based on said load signal and said kill signal value output by said kill queue.

Roth et al. teaches disclose logic, coupled to said kill queue, for generating said valid signal during the second clock cycle based on said load signal and said kill signal value output by said kill queue (the valid bit it based on the kill signal because they are the same value; the kill signal is based on the load signal because the entry outputted by the queue is dependant on the pointer; see above regarding claim 3).

It would have been obvious for one of ordinary skill in the art at the time of the invention to have modified the invention of Kroesche by adding logic, coupled to said kill

queue, for generating said valid signal during the second clock cycle based on said load signal and said kill signal value output by said kill queue, as taught by Roth et al., in order to keep track of valid values for instructions after the instruction queue (Note that if a valid signal based on the kill signal was no outputted from the queue, there would be no way to keep track of which instructions were valid).

25. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cousin (U.S. Patent No. 6,725,357 B1) in view of Roth et al. (U.S. Patent No. 6,754,808 B1).

26. Regarding claim 16, Cousin discloses an apparatus for killing an instruction loaded into an instruction queue (fig. 1, prefetch buffer 6; instruction is loaded from program memory; col. 3, lines 28-32) of a microprocessor during a first clock cycle (when instruction is loaded into the prefetch buffer) and output from a bottom entry of the instruction queue (prefetch buffer) during a second clock cycle (clock cycle when instruction is outputted to decoder), subsequent to the first clock cycle (when the queue is not empty, the instruction will sit in the queue for at least 1 clock cycle; col. 6, 35-38) and a load signal (pointer; col. 7, lines 35-40) [apparatus of claim 1], wherein the instruction is output from the bottom entry of the instruction queue during the second clock cycle to an instruction translator (decoder 8 and μ instruction generator 10; see col. 3, lines 31-33) for translating the instruction into one or more microinstructions to be executed by the microprocessor (col. 3, lines 33-35).

Cousin does not explicitly disclose a kill signal, for conveying a value generated during a third clock cycle subsequent to the first clock cycle; a kill queue, coupled to

said kill signal, for loading said kill signal value generated during said third clock cycle, and for outputting said kill signal value during the second clock cycle; and a valid signal, coupled to said kill queue, generated during the second clock cycle for indicating whether the instruction is to be executed by the microprocessor, wherein said valid signal is false if said kill signal value output by said kill queue during the second clock cycle is true or that the signal is coupled to said kill queue and that the microinstructions are selectively executed based on a based on said valid signal.

Roth et al. teaches that the microinstructions are selectively executed based on said valid signal (col. 4, lines 3-6; writing back a microinstruction is considered to be executing that microinstruction).

See above regarding claim 4 for all other limitations.

For the invention to be successful, the microinstructions would have to inherit the kill signal value of the instructions they are based on.

See above regarding claim 4 for all other limitations.

It would have been obvious for one of ordinary skill in the art at the time of the invention to have modified the invention of Cousin by executing microinstructions selectively based on said valid signal, as taught by Roth et al., in order to easily keep track of invalid microinstructions (Roth et al., col. 3, lines 45-56; see above regarding claim 4).

27. Claims 14, 22- 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Roth et al. (U.S. Patent No. 6,754,808 B1) in view of Green (U.S. Patent No. 6,041,405).

28. Regarding claim 14, Roth et al. discloses the apparatus of claim 13 (see above regarding claim 13).

Roth et al. does not explicitly disclose said variable length instruction comprises an x86 instruction.

Green teaches said variable length instruction comprises an x86 instruction (col. 1, lines 15-19).

It would have been obvious for one of ordinary skill in the art at the time of the invention to have modified the invention of Roth et al. by changing said variable length instruction to comprise an x86 instruction, as taught by Green, in order to increase compatibility with existing software because the x86 instruction set is widely accepted (col. 1, lines 15-19).

29. Regarding claim 22, Roth et al. disclose the method of claim 17 (see above regarding claim 17), further comprising: predicting said instruction is a taken branch instruction (col. 3, lines 52-56; branch prediction predicts branches are taken).

Roth et al. does not expressly disclose said predicting occurs prior to said loading said instruction into said first queue; detecting a misprediction of said branch instruction; and said generating said kill signal during said second clock cycle in response to said detecting said misprediction.

Green teaches predicting prior to said loading said instruction into said first queue (fig. 2, Branch Prediction Unit 14; col. 5. lines 36-41); detecting a misprediction of said branch instruction (col. 9, lines 36-40); and said generating said kill signal during said second clock cycle in response to said detecting said misprediction (col. 9, lines 38-44; when a branch is mispredicted, it is flushed or killed).

The invention of Roth et al. discloses that it does contain a branch predictor, but does not explicitly say how the predictor works. A typical branch predictor detects if a branch is mispredicted, and kills instructions which should not have been executed. The teachings of Green as noted above would have easily been added to the invention of Roth et al. with little change.

It would have been obvious for one of ordinary skill in the art at the time of the invention to have modified the invention of Roth et al. by predicting prior to said loading said instruction into said first queue, as taught by Green, in order to generate branch prediction information as early as possible; detecting a misprediction of said branch instruction, as taught by Green, in order to ensure correct program flow (if branch predictions were never verified, program flow would be incorrect); and said generating said kill signal during said second clock cycle in response to said detecting said misprediction, as taught by Green, in order to ensure invalid instructions are not executed.

30. Regarding claim 23, Roth et al./Green discloses the method of claim 22 (see above regarding claim 22).

Roth et al. does not expressly disclose that a branch target address cache of the microprocessor performs said predicting said instruction is a taken branch instruction.

Green teaches using a branch target address cache (fig. 2, Branch Prediction Unit 14) of the microprocessor performs said predicting said instruction is a taken branch instruction (col. 6, lines 34-40).

It would have been obvious for one of ordinary skill in the art at the time of the invention to have modified the invention of Roth et al. by having a branch target address cache of the microprocessor performing said predicting said instruction is a taken branch instruction, as taught by Green, in order to determine fetch addresses and fetch instructions before the address of the instruction would normally be known (Green, col. 6, lines 34-36).

31. Regarding claim 24, Roth et al./Green discloses the method of claim 22 (see above regarding claim 22).

Roth et al. does not expressly disclose said misprediction of said branch instruction comprises a misprediction of a length of said branch instruction.

Green teaches said misprediction of said branch instruction comprises a misprediction of a length of said branch instruction (col. 13, lines 46-49).

The length predictor would operate at the same time as the branch predictor. If a length were to be mispredicted, the instructions that were speculatively decoded would have to be killed and the correct instructions would be decoded non-speculatively.

It would have been obvious for one of ordinary skill in the art at the time of the invention to have modified the invention of Roth et al. by having said misprediction of said branch instruction comprise a misprediction of a length of said branch instruction in order to reduce the time that instruction alignment unit must wait for instruction length (Green, col. 13, lines 5-7).

32. Regarding claim 25, Roth et al./Green discloses the method of claim 22, wherein said misprediction of said branch instruction comprises a misprediction of an address of said branch instruction (see above regarding claim 22; if the processor predicts addresses of branch instructions, it must be incorrect some of the time).

Regarding claim 26, Roth et al./Green discloses the method of claim 22, wherein said misprediction of said branch instruction comprises said branch instruction being a non-branch instruction (functional processors must be able to handle mispredicting non-branch instructions as being branch instructions; N*ote that Black [U.S. Patent No. 5,761,723] states that "translation arrays may erroneously map a non-branch instruction to a branch instruction in the BTAC" [col. 1 last line, col. 2, lines 1-2]).*

33. Claims 27 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Roth et al. (U.S. Patent No. 6,754,808 B1) in view of Green (U.S. Patent No. 6,041,405) and Fite (U.S. Patent No. 5,142,634).

34. Regarding claim 27, Roth et al./Green discloses the method of claim 17 (see above regarding claim 17), further comprising: branching the microprocessor based on a prediction that a branch instruction is taken (when branch prediction occurs, it is implied that the branch is taken before it is known), wherein said instruction is sequential to said branch instruction (all instructions enter the pipeline, therefore the instruction after the branch must enter).

Roth et al. does not expressly disclose that said generating said kill signal is during said second clock cycle after said branching the microprocessor.

Fite teaches generating said kill signal during said second clock cycle after said branching the microprocessor (steps 44-46; see fig. 3, col. 13, lines 57-60).

It would have been obvious for one of ordinary skill in the art at the time of the invention to have modified the invention of Roth et al. by generating said kill signal during said second clock cycle after said branching the microprocessor, as taught by Fite, in order to decrease the number of instructions executed by executing only instructions that are predicted to be needed. This saves power.

35. Regarding claim 28, Roth et al./Green discloses the method of claim 17, wherein said instruction sequentially follows a branch instruction predicted taken (see above regarding claim 27).

Roth et al. does not expressly disclose generating said kill signal during said second clock cycle in response to detecting said branch instruction is predicted taken.

Fite teaches generating said kill signal during said second clock cycle in response to detecting said branch instruction is predicted taken (steps 44-46; see fig. 3, col. 13, lines 57-60).

Regarding motivation, see above regarding claim 27.

Allowable Subject Matter

36. Claims 5 and 6 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject matter: Claims 5 and 6 recite limitations which force said load signal to be a single bit (true or false). A single bit indicating whether an entry is loaded into the bottom of a queue in a previous clock cycle could not be found.

Conclusion

37. The following is text cited from 37 CFR 1.11(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the

objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jesse R. Moll whose telephone number is (571)272-2703. The examiner can normally be reached on M-F 8:00 am - 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Huynh can be reached on (571)272-4147. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JM 2/15/06

HENRY W. H. TSAI
PRIMARY EXAMINER